charles river analytics

Charles River Analytics Inc.
625 Mount Auburn St.
Cambridge, MA 02138
www.cra.com

p: 617.491.3474
f: 617.868.0780

15 DEC 2009

Reference: Government Contract No. N00014-09-C-0050, “Enhancing Simulation-based Training Adversary Tactics via Evolution (ESTATE)”
Charles River Analytics Contract No. C08098

Subject: Contractor’s Status Report: Quarterly Status Report #4
Reporting Dates: 09/15/2009 – 12/15/2009

Dear Dr. Hawkins,

The following is the Contractor’s Quarterly Status Report for the subject contract for the indicated period. During this reporting period work has concentrated on Task 4: Develop Trainee Model Processing.

1. Summary of Progress

During this reporting period, we focused on working on mining the existing MoneyBee data from our academic partner, Brandeis University, to examine student performance and began to construct a way to create a representation of strategies for future research.

1.1 Exploring the MoneyBee Dataset

The goal of this task was to discover if a method exists to use a significant amount of data to measure the students’ changing performances and to determine if students are gaining proficiency in this pre-algebra activity. This analysis is in anticipation of future examinations of item response curves and strategy choices.

MoneyBee is a coin algebra activity. The student is given a sum and a number of coins and has to pick out which coins add up to the sum. A session consists of five paired exercises. In each exercise, students create problems for the other to solve followed by the reception of a student-created problem and a graphical workbench for solving the problem. Each exercise collects a detailed timeline, down to a tenth of a second, recording when players add and subtract coins towards solving the problem they are presented.

Figure 1-1 through Figure 1-3 show the game history of three Moneybee players. Each line is a graphic representation of a player’s actions during a single game. Blue dots indicate when a player adds or subtracts a quarter, green is for dimes, red is for nickels and yellow is for pennies. The x-axis represents time in tenths of seconds, and the y-axis is the accumulated value in the player’s stack of coins.

Report Documentation Page

Form Approved
OMB No. 0704-0188

1. REPORT DATE **2009**	2. REPORT TYPE	3. DATES COVERED **00-00-2009 to 00-00-2009**
4. TITLE AND SUBTITLE **Enhancing Simulation-Based Training Adversary Tactics vs. Evolution (ESTATE)**		5a. CONTRACT NUMBER 5b. GRANT NUMBER 5c. PROGRAM ELEMENT NUMBER
6. AUTHOR(S)		5d. PROJECT NUMBER 5e. TASK NUMBER 5f. WORK UNIT NUMBER
7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES) **Charles River Analytics, Inc,625 Mount Auburn St,Cambridge,MA,02138**		8. PERFORMING ORGANIZATION REPORT NUMBER
9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)		10. SPONSOR/MONITOR'S ACRONYM(S) 11. SPONSOR/MONITOR'S REPORT NUMBER(S)
12. DISTRIBUTION/AVAILABILITY STATEMENT **Approved for public release; distribution unlimited**		
13. SUPPLEMENTARY NOTES		
14. ABSTRACT		
15. SUBJECT TERMS		

16. SECURITY CLASSIFICATION OF:			17. LIMITATION OF ABSTRACT	18. NUMBER OF PAGES	19a. NAME OF RESPONSIBLE PERSON
a. REPORT **unclassified**	b. ABSTRACT **unclassified**	c. THIS PAGE **unclassified**	**Same as Report (SAR)**	**10**	

Standard Form 298 (Rev. 8-98)
Prescribed by ANSI Std Z39-18

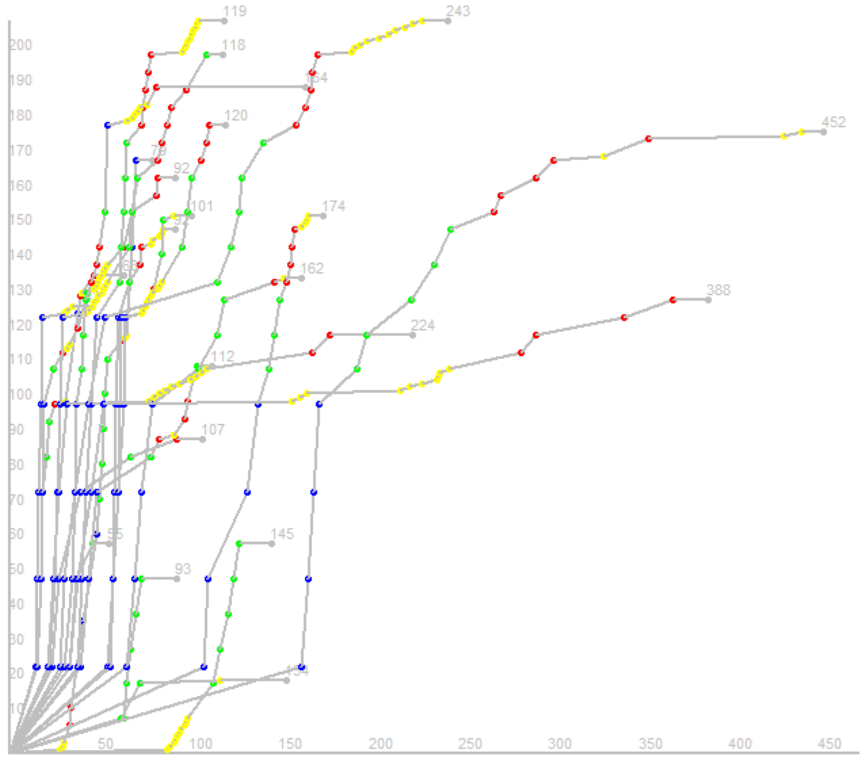


Figure 1-1: Play history for '1b2b3b'

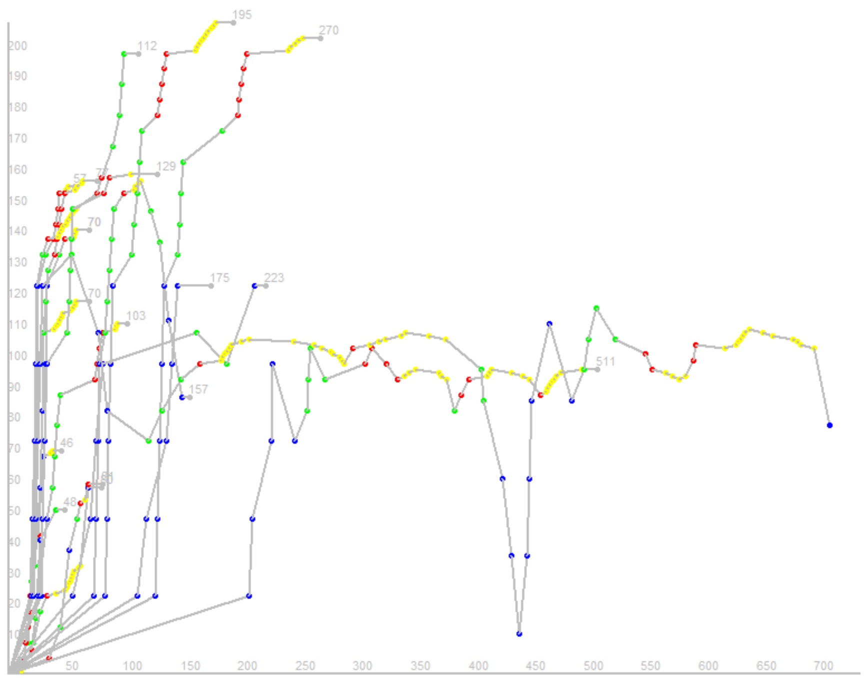


Figure 1-2: Play history for 'eminem05'

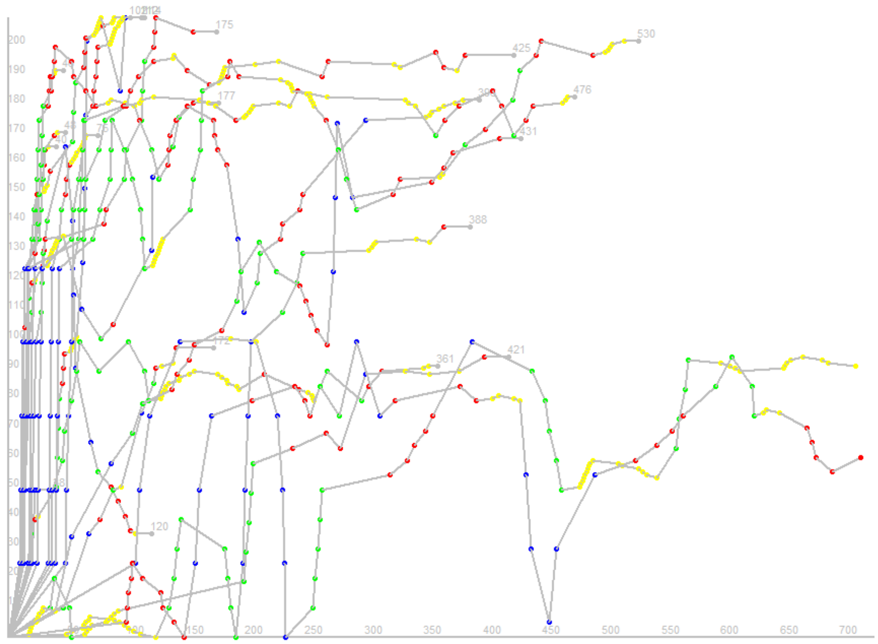


Figure 1-3: Play history for '1soccergal'

'1b2b3b' appears to be a consistent player, ‘eminem05’ appears to be a less consistent player with 2 erratic answers and ‘1soccergal’ appears to be a player with many erratic answers who has not yet formed a consistent strategy.

Similarly, the challenges themselves can be graphed to gauge the trends in players’ strategies. Figure 1-4 depicts the history of challenge 1218, making 78 cents with 8 coins. For instance, on this challenge it appears that most players begin by adding quarters, and then move to smaller valued coins.

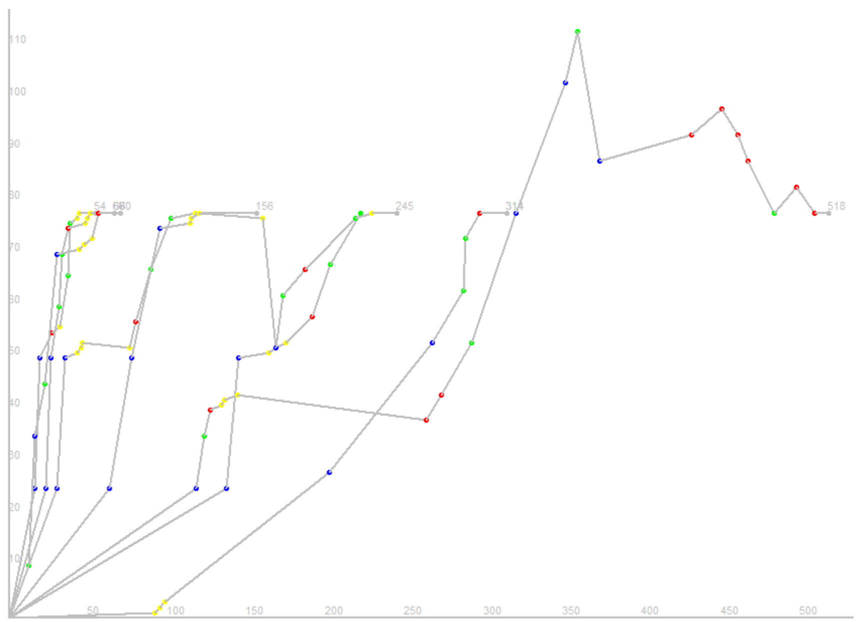


Figure 1-4: Challenge 1218, 78 cents with 8 coins

Moneybee is played in sessions which consist of eight games. Our first task is to determine if players are performing better in later sessions.

Figure 1-5 illustrates the percentage of players performing better than average over the course of several sessions. The blue line represents players' first session. The x-axis is the game number within that session and the y-axis is the percentage of that group above the mean (outliers in the dataset cause most players to outperform the mean). The percentage value of the blue line at y-value 1 indicates that 47% of players do better in their first game of their first session than the mean and the blue line at 2 indicates that 58% of players do better in their second game of their first session than the mean. Within each session, players appear to improve and appear more proficient in later sessions; game plays in session 2 (orange line) have a higher % than the first session (blue line), game plays in session 3 (yellow) are slightly better than in session 2, and game plays in session 4 (green) are slightly better than in session 3.

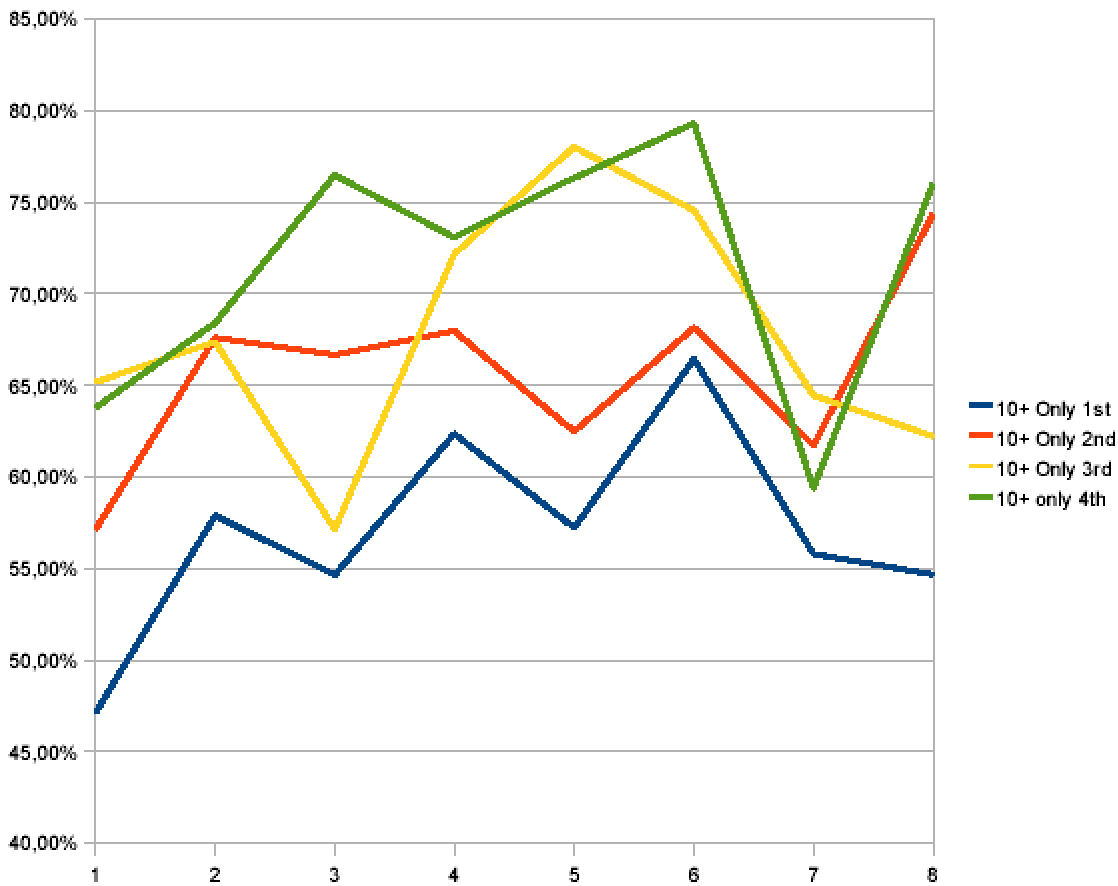


Figure 1-5: Percentage above mean by game number, split by play sessions.

One explanation for this difference may be attrition. Players that are not doing well at the game may drop out, leaving only the players that are better than average. To examine this possibility, Figure 1-6 graphs only the data from players who completed all four sessions. The trends appear the same. Players seem to improve between sessions, and hence attrition does not appear to be the cause of the improvement.

In conclusion, this examination of the dataset indicates that students are becoming more proficient at the coin algebra task over game plays. They are developing strategies and learning between sessions.

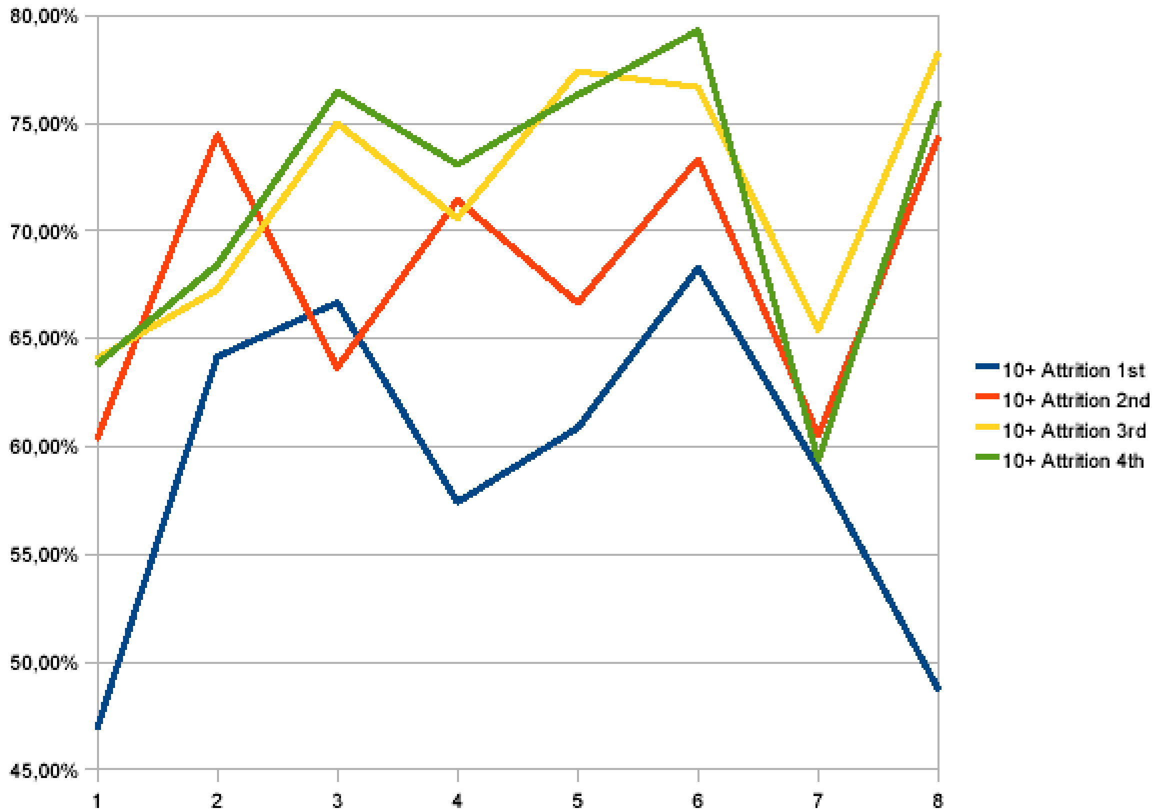


Figure 1-6: Percentage above mean by game number, split by play sessions. Only players that completed four or more sessions.

1.2 Constructing a Compact Representation of MoneyBee State-Action Strategies

In the MoneyBee game, players are given a set of coins and a task to assemble *n* cents with *m* coins. To assess the players' strategies in the MoneyBee game, a compact representation was developed that would allow strategies to be expressed in such a way that they may be compared. Actions from the same state in the game are collected together for analysis. A single strategy choice point is illustrated in the following figure.

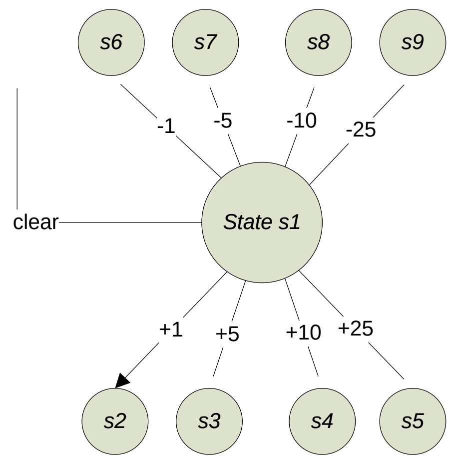


Figure 1-7: MoneyBee State-Transition Representation

A MoneyBee state consists of:

- Number of coins needed
- Number of cents needed
- Number of quarters remaining
- Number of dimes remaining
- Number of nickels remaining
- Number of pennies remaining

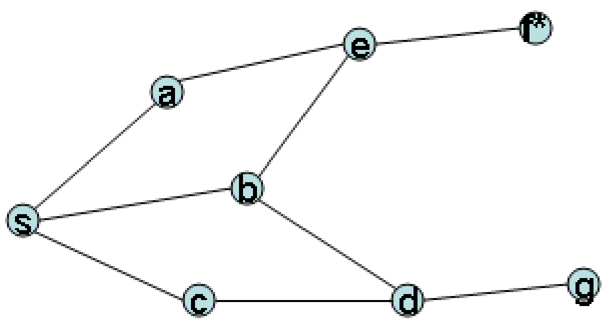
The above model has been implemented in python. The MoneyBee database recording all of the player moves has been parsed and converted into this format for analysis. Like actions are collected into states, and choices by the players for each state (whether they add a quarter or subtract a nickel, for instance) are tallied. We are currently using this representation for a more sophisticated analysis about player performance and strategy development. Lessons learned from this analysis can then be applied in a general-purpose fashion to other challenge / response games.

1.3 Strategy Space Approach

To enhance our representation of player's strategies beyond that of a simple skill value, we examine the use of strategy representations in a discrete space. This approach will allow the modeling of trainee strategy evolution over time, instead of only trainee skill level. ESTATE will be able to use this information to target weaknesses in strategy when constructing challenges.

1.3.1 State Space Representation

This problem formulation can be solved by employing a conditional planner to select the sequence of challenges. Consider this simple strategy space:

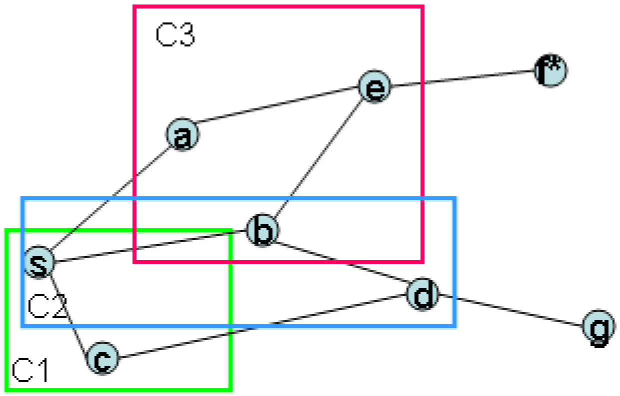


The trainee begins using the strategy **s** and may modify that strategy to either a, b, or c upon failing a challenge. In this example, f* is the goal strategy. It is desirable for us to guide the trainee to adopt this strategy. The system has the following set of challenges to select from (each challenge is listed with the set of strategies it will defeat):

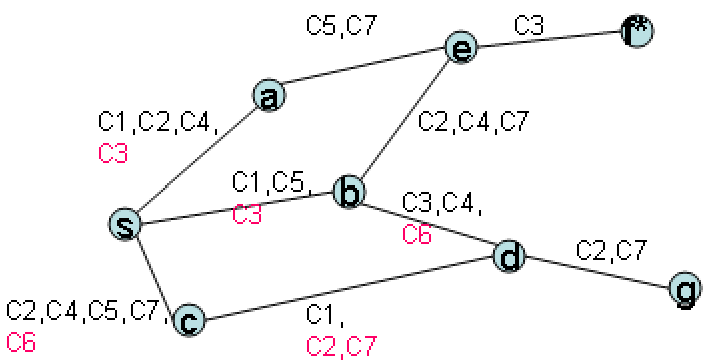
- C1 – s, c
- C2 – s, b, d
- C3 – b, e, a
- C4 – s, b
- C5 – s, a
- C6 – c, d, g
- C7 – s, a, b, d

We assume that choosing a challenge for a trainee forces the trainee to adopt a connected strategy (one that is reachable within the local neighborhood) that is not defeated by the challenge.

For example, the system may force the trainee into the goal state by choosing the sequence C1, C2, C3. C1 forces the trainee to move to either a or b. If the trainee moves to strategy b, C2 forces him to move to e. Then C3 forces the move to f*. If the trainee goes from s to a, however, then C2 would be the wrong choice. C2 does not force a change in strategy from a, and following on with C3 may push the trainee back to s.



The strategy network and accompanying challenge set can be combined to record which challenges enable which transitions. The transition s→a is enabled by C1, C2, and C4, and the transition a→s is enabled by C3.



This visualization suggests a path planning solution. Specifically, this problem addresses path planning with uncertain actions; choosing C1 in state s will result in either a or b. A conditional planner may be employed to create conditional plans of challenge choices. See Draper, et al.[1], and Bertoli, et al.[2] for examples of conditional planners. Note that this is also an area studied by partially observable Markov decision processes (POMDPs). Applying such an approach raises questions of scalability, which is an area of focus for future work.

---

[1] Denise Draper, Steve Hanks and D. Weld, "Probabilistic Planning with Information Gathering and Contingent Execution," Proceedings of AIPS-94, June 1994

[2] P. Bertoli, A. Cimatti, M. Roveri and P. Traverso "Planning in Nondeterministic Domains Under Partial Observability via Symbolic Model Checking" IJCAI-2001 (Seventeenth International Joint Conference on Artificial Intelligence). Seattle, Washington, USA, August 4th - 10th, 2001

1.3.2 State/Action Strategy Representation

Instead of strategies that are black boxes (i.e., only nodes within a strategy space), strategies may be represented as sets of state/action pairs as in extensive form game theoretic analysis. In this representation, a strategy for tic-tac-toe may state that *if* the board is empty *then* mark an ‘o’ in the middle, and *if* the board has an ‘o’ in the middle and an ‘x’ in the top left *then* mark an ‘o’ in the top middle, and so on. A complete strategy defines an action for every game state.

We begin with the following assumptions:

1. **Single player, discrete game.** There is no opponent or adversary changing the game state between moves.
2. **Actions are deterministic.** Actions define a transition to one and only one game state
3. **Game dynamics are constant** and not conditioned on a challenge. With the above assumption, choosing an action *a* in state *s0* will always result in state *s1*, regardless of the challenge. The only reason the state changes is due to the player’s action.
4. **Goals are constant**. The object of the game is to obtain one of a set of goal states defined by a payoff function G: state → real. We restrict the set of states that have non-zero payoff to be terminal states (no points accrued during play).

A game is defined as a set of *dynamics* (state, action, new state triplets <s,a,s’>) and a *goal state function*. A **challenge** within a game is an initial state. The challenge defines which strategies are successful, those that lead from the initial challenge state to a goal state, and which are not, those that do not lead to a goal state. Thus, any non-terminal state is a possible challenge.

1.3.2.1 Constructing Challenges

Given a complete specification of a strategy to defeat (perhaps a perceived strategy of a trainee), a challenge that will defeat that strategy can be obtained by searching through the state/action space. In the simplest case, we select a challenge that will cause the strategy to fail in one move:

1. Select an action <s,a,s’> from the strategy S such that s’ is a losing state
2. s is a challenge that will defeat S

Backward-chaining can be used to find states that will result in the strategy failing after more than one move

1. Select an action <s,a,s’> from the strategy S such that s’ is a losing state
2. s_c = s
3. Loop:
    a. find s_prev | <s_prev, a, s_c> is in S
    b. s_prev is a possible solution
    c. s_c = s_prev

1.3.2.2 Partial Strategies and Uncertainty

Given a partial specification for a strategy (in which some state/action pairs are unknown), a probability value for the likelihood that the strategy will fail at a challenge may be computed. At any state, the probability of the strategy failing at that state is the sum of the probabilities of failing at neighbor states times the probability that those states will be chosen.

P_loss(s,S) = ∑ P_loss(s',S) * P_choose(s',S)

For known state/actions in the strategy, P_choose(s') is 1, and for unknown states, P_choose(s') may be the inverse of the number of possible actions in s (1/n). For losing states, P_loss(s',S) is 1, and for winning states, P_loss(s',S) is 0. Challenges that are most-likely to defeat the partial strategy may be discovered by employing an A* search beginning from the loss states using P_loss(s,S) as the admissible heuristic.

2. Scheduled Items

During the next reporting period, we plan to focus on the following tasks:

- Revising Item Response Graphs
  - A posteriori examination of data instead of in-progress examination
  - Partitioning of data for individual users over time
- Mining MoneyBee Data using a state-transition representation
  - Do players in similar game states make similar decisions?
  - Do players employ consistent strategies?
  - Do players keep successful strategies and discard unsuccessful ones?
- Continuing to explore State-Action Strategy representation
  - Re-examine competitive pathologies in coevolution as applied to this representation
  - Examine challenge construction that avoids competitive pathologies

Sincerely,

Brad Rosenberg
Principal Investigator